Aliaksandr Z.
Expert System Software Developer, Security Architect

SUMMARY

- Experienced System Software Engineer with 12+ years of experience designing and implementing firmware-level and system-level security solutions, sophisticated OS-kernel extensions and device drivers, hi-loaded systems development, and data processing. - Solid foundation in building secure, real-time operating systems and extending Linux kernel capabilities, with strong proficiency in both C and C++. - Instrumental in architecting and implementing UEFI hypervisors and firmware for Class B medical devices, emphasizing security features like CryptoCell for secure communications. - Proven capabilities in vulnerability analysis, leveraging tools like Angr and AFL for automated firmware testing. - Played a key role in SDLC, driving modern software architecture practices and security foundations in projects across medical and information security industries. - Upper-Intermediate English

TECHNICAL SKILLS

Main Technical Skills	C/C++/C# (10 yr.), Linux (5 yr.)
Programming Languages	Aarch32_64, GLSL, PHP, Python (5 yr.), x86 Assembly
C++ Libraries and Tools	C/C++/C# (10 yr.)
.NET Platform	Identity Server
JavaScript Frameworks	Node.js
AI & Machine Learning	OCR, TensorFlow Serving
Data Analysis and Visualization Technologies	Celonis
Security	GPO (Group Policy Object)
Databases & Management Systems / ORM	MariaDB
UI Frameworks, Libraries, and Browsers	Gstreamer
Amazon Web Services	AWS Security Groups (5 yr.)
Google Cloud Platform	GCE
QA, Test Automation, Security	AFL Service Solutions, Angr, TrustZone
Web/App Servers, Middleware	Apache HTTP Server, XAMPP (X, Apache, MariaDB, PHP, Perl)

Deployment, CI/CD & Administration	CI/CD
Third Party Tools / IDEs / SDK / Services	CMake
Codecs & Media Containers	Ffmpeg
Virtualization, Containers and Orchestration	KVM (for Kernel-based Virtual Machine) (5 yr.), Qemu
Operating Systems	Linux (5 yr.), Windows (5 yr.)
Methodologies, Paradigms and Patterns	REST, RPC (Remote Procedure Call)
SDK / API and Integrations	TPM API, Windows API
Other Technical Skills	BCM, CMSIS, eBFP, Firmware Development, FreeRTOS, HAL, IMPI, OP-TEE, Perl), QSYM, Rv32_64, SBI, XVisor, ZephyrRTOS

WORK EXPERIENCE

Senior Embedded Software Engineer - FW Project

Mar 2023 - Now

Summary: Implementation and architecture design of firmware for Class B medical devices, focusing on RTOS-based firmware, security, and automated firmware testing.

Responsibilities:

- Headed FW part of the project
- Architected SW part of the new brand device based on old legacy one
- Designed and implemented RTOS-based firmware in C++ (17-20), leveraging modern approaches for Interfaces, CMSIS, HAL, State Machines, Event-Based architecture, as well as modern C++ (17-20) SW architecture approaches
- Designed and implemented security foundations and extensions based on CryptoCell functionality, devoted to secured BLE/USB communications, encrypted storage
- Designed and established automated FW testing leveraging CI/CD pipelines and FW emulation/re-hosting

Technologies: C++, CMSIS, HAL, State Machines, Event-Based architecture, CryptoCell, BLE, USB, CI/CD.

Senior Embedded Software Engineer - Automated Discovery Firmware Vulnerabilities Project

Mar 2021 - Mar 2023

Summary: Leadership and architecture of a project for automated discovery of firmware vulnerabilities leveraging advanced analysis techniques.

Responsibilities:

- Headed project and designed its architecture and integrations into the main product
- Leveraged symbolic-execution and dynamic binary instrumentation for automated discovery of vulnerabilities in firmware binaries
- Leveraged the existing state-of-the-art fuzzing and emulating techniques to automated firmware analysis to explore software weaknesses and detect exploitable vulnerabilities
- Implemented lightweight software-based API-level emulation for mbedOS, FreeRTOS, thereby sufficiently improves the speed and coverage of firmware emulation and analysis
- Adopted OCR and image processing approaches for detecting functions signatures in stripped binaries, which made it possible to achieve high detection accuracy

Technologies: Symbolic execution, dynamic binary instrumentation, mbedOS, FreeRTOS, OCR, image processing.

Senior Information Security Software Engineer - Afina Systems

Jan 2019 - Mar 2021

Responsibilities:

UEFI Hypervisor Project:

- Architected, designed, and implemented UEFI hypervisor for Intel VT-x/dbased chipsets
- Enhanced functionality of existing UEFI DXE drivers to support more UEFI protocols

Devices Restricted Space Project:

- Architected, designed, and implemented framework (host and guest parts) for fine-grained policy-based control of guest devices in a host-maintained pace for TrustZone-enabled and MultiZone-enabled SoCs
- Developed integrations of the framework into ZephyrRTOS and FreeRTOS, developed Linux kernel integration module

Linux Desktop Management Project:

- Headed development of Desktop Management project on Linux
- Designed core project architecture, and interviewed applicants on project
- Implemented eBFP-based activity monitor to detect anomalous behavior in user's applications
- Implemented various Linux Security Module extensions for managing desktop users activities

Technologies: UEFI hypervisor, Intel VT-x, TrustZone, ZephyrRTOS, Linux, eBFP.

Senior Software Development Engineer - Falcongaze Company

Aug 2014 - Jan 2019

Summary: Development of security frameworks for data prevention, removable device control, and printer job monitoring in a system software environment.

Responsibilities:

Data Lost Prevention Project :

- Designed and implemented file-systems control management and context filtering framework, which provides the ability to shadow file operations and data for further analysis, and on-the-fly context-based data access control for users
- Introduced and adopted C++ usage in existing Windows kernel modules, that improved code quality and facilitated maintenance of existing projectsDevice

Access Control Project :

- Designed and implemented removable devices access control framework, which allowed policy-based runtime inserting/removing from station suspicious devices backed by various interfaces (USB, PCIe, Bluetooth, SATA, IDE, HDMI)
- Implemented policy-based time-restricted and activity-restricted access to the MTD-devices

Data Processing Server Project :

- Designed and implemented cloud-fashioned architecture for images and videos storage and processing
- Implemented PKI, Authentication and Authorization protocols, Cache Policy and Load Balancing, exposed RPC and REST interfaces
- Leveraged image and video streaming and frameworks such as GStreamer, FFmpeg, Tesseract OCR, Abby OCR

Printer Subsystem Management Project :

- Developed and maintained printer usage control framework
- Leveraged Windows Device Management Subsystem on both user-level applications and kernel-level modules for policy-based printers access control
- Implemented shadowing of printer jobs for further analysis
- Implemented context-based restrictions for printed documents

Technologies: C++, Windows kernel modules, GStreamer, FFmpeg, REST, RPC.

Middle Software Developer C/C++ - Security Software Systems Inc

Aug 2011 - Aug 2014

Summary: Development of network filters, traffic interceptors, and application control systems for Windows network stack.

Responsibilities:

Network Packet Filtering Engine Project :

- Implemented parser for proprietary network protocols (MAPI, YAHOOP, MTPRCP, XSRTP)

- Designed and implemented the application's network functions hooking library, which allowed to intercept and parse many closed and obfuscated messaging protocols (WhatsApp, Skype)

Network Traffic Interceptor Project:

- Designed and implemented Windows WFP kernel module and control daemon for interception and manipulation of TCP/UDP packets in Windows 8+ network stack, which replaced old existing TDI-based solution and provided more robust and flexible functionality on modern Windows releasesApplications

Control Engine Project:

- Designed and implemented Windows Printing Subsystem shadowing and context controlling library, which provides creating copies of printed data and allows/rejecting printer operations based on data context

Technologies: C/C++, Windows WFP, TDI, Printing Subsystem.

Junior System Software Engineer - VBA32 Ltd, MINSK

Aug 2009 - Aug 2011

Summary: x86 CPU emulator development and Windows environment emulator enhancements for improved performance and malicious code detection.

Responsibilities:

x86 CPU Emulator Project:

- Implemented AMD-specific instruction set (MONITORX, MCOMMIT, INVLPGB, etc)
- Implemented some undocumented x86 instructions from Ralf Brown's List

win32 Environment Emulator Project :

- Designed and implemented Win32 Registry Subsystem emulation, that provide the ability to detect sophisticated malicious code
- Moved the entire project from using Squirrel to Lua, which allowed an increased performance of emulation and decreased time complexity for adding new futures into the emulator

Technologies: x86 instruction set, Ralf Brown's List, Win32 Registry, Squirrel, Lua.

EDUCATION

Belarusian State University of Informatics and Radioelectronics

Engineer of Radioinformatics Systems, Master of Engineering Science in Telecommunications and Radio Informatics